Table of Contents

1. Introduction....3
  1.1. Overview....3
  1.2. Package Contents....3
  1.3. Disclaimer....3
  1.4. EMC Conformance Statement....4
  1.5. System Requirements....4
2. Warranty....5
  2.1. Technical Support....5
3. Quick Start Guide....6
4. Feature Descriptions....7
  4.1. Digital I/O Ports....7
  4.2. Analog Channels....8
  4.3. Event Monitoring....11
  4.4. Pulse Width Modulation (PWM) Outputs....11
  4.5. Digital Counters....11
  4.6. Built-in LEDs....12
  4.7. EEPROM Storage....12
  4.8. Power Up Defaults....12
  4.9. TCP/IP Connections....13
5. Software Installation....14
  5.1. Windows Installation....14
  5.2. Linux Installation....14
6. Network Connections and Configuration....15
  6.1. Connecting the ETH32 to the Network....15
  6.2. Configuring the ETH32....15
7. Developing PC Applications....20
8. Hardware Information....21
  8.1. Dimensions and Connector Locations....21
  8.2. Connector Pinouts....22
  8.3. Power Options....26
  8.4. I/O Ports....27
9. Specifications....29
  9.1. Absolute Maximum Ratings....29
  9.2. Electrical Characteristics....30
  9.3. Performance Characteristics....31
  9.4. TCP/IP Connections....35
  9.5. Analog to Digital Converter....35

9.6. Counters....36
9.7. Pulse Width Modulation Outputs....36
10. Appendix: ETH32 Accessories....36
10.1. Power Supply....36
10.2. Breakout Boards....36
10.3. Breadboard Adapters....39
10.4. Cables, Connectors, Adapters....39
10.5. Relay Boards....40
10.6. Wireless Adapter....41

1. Introduction

1.1. Overview

The ETH32 is a flexible, general purpose I/O device which communicates over Ethernet. It features 34 configurable I/O lines which can be configured to operate in various modes. The available modes include digital I/O, analog inputs, PWM outputs, and counter inputs. With this variety of modes available, the ETH32 is capable of monitoring and controlling many different processes and applications.

A notable feature of the ETH32 is its 'event-driven' capability when handling inputs. When enabled, this feature eliminates the need to poll inputs by locally monitoring input(s) on the ETH32 and sending an event notification via Ethernet when an input meets certain criteria. Event notification capability is included for digital inputs, analog inputs, and counters.

In order to support future enhancements, the ETH32 supports firmware upgrades in the field. Additional programming hardware (not included) is required for this function.

1.2. Package Contents

The package you receive should contain the following items:

1. This manual
2. The ETH32 device
3. CD containing application libraries, sample code, and documentation

1.3. Disclaimer

The ETH32 is not intended or designed for military, medical, or other life-critical applications. Winford Engineering shall not be held responsible or liable for damages or injury that occur as a result of the use of this product.

1.4. EMC Conformance Statement

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

-Reorient or relocate the receiving antenna.
-Increase the separation between the equipment and receiver.
-Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
-Consult the dealer or an experienced radio / TV technician for help.

Changes or modifications to the device not expressly approved by Winford Engineering could result in EMC non-conformance, thereby voiding the user's authority to operate the device.

1.5. System Requirements

The programming libraries, example applications, and configuration utility provided by Winford Engineering require a PC with the following minimum requirements:

- Intel 32-bit or 64-bit* CPU compatible architecture
- Windows 95 or later (including Vista and Windows 7), or Linux distribution with 2.0 kernel or later
- Properly configured Ethernet adapter
- Properly configured TCP/IP configuration and IP address
- To create PC applications utilizing the ETH32 API, one of the programming environments / compilers listed in the *Developing PC Applications* section on page 20 is required.

Using the ETH32 Protocol Reference document, other platforms and programming languages can be used.

* Pre-compiled libraries for 64-bit systems are currently provided for Windows platforms. The API source code can be used to compile 64-bit libraries for Linux, or potentially for other non-Intel systems as well.

2. Warranty

Winford Engineering provides a 1-year warranty for the ETH32 device. This warranty does not cover the misuse or abuse of this product. In the event of product failure, Winford Engineering should be contacted for information regarding the return and replacement of the product.

2.1. Technical Support

If you find yourself in need of technical assistance beyond what is covered in this manual, there are several ways to contact Winford Engineering. If your problem is not urgent, we would prefer that you use e-mail. Please provide us with your preferred e-mail address, as well as a phone number where you wish to be reached. This will help us get back to you in a timely fashion.

Technical Support Contacts

Phone: 1-989-671-9721
FAX: 1-989-671-2941
E-mail: support@winford.com

Winford Engineering
Technical Support
4169 Four Mile Road
Bay City, MI 48706

3. Quick Start Guide

If you are anxious to get up and running quickly, then this section is for you. This section explains how to connect the board, configure it properly, and run a simple demo program.

1. **PC Software**
   Install the libraries and utilities from the product CD onto your computer if you haven't already. Simply run the `eth32_install.exe` file on the CD and select all available items to be installed. You may see the *Software Installation* section on page 14 for further information.

2. **Connections**
   For this simple demo, only two connections are needed:

   - Connect the RJ45 ethernet jack to your ethernet network. The ETH32 may be connected to a 10 Mbit/second port (10BaseT) or a 10/100/1000 Mbit/sec port.
   - Connect power.

3. **ETH32 Configuration**
   The ETH32 now ships with DHCP enabled. If your network has a working DHCP server, the ETH32 will automatically be assigned a valid IP address for your network. Otherwise, you can manually configure the ETH32 with IP address settings. To do that, be sure the ETH32 is connected to the same network as your PC and run the ETH32 network configuration application. This is found at *Winford ETH32 ➔ ETH32 Network Configuration* on your Start Menu. Assign the ETH32 an IP address that is available and valid on your network. See the *Network Connections and Configuration* section on page 15 for more information.

4. **Run a Demo Program**

   - Run the LEDs example application. This is found at *Winford ETH32 ➔ Examples ➔ LEDs Example (VB6)* or *LEDs Example (C#.NET)*
     on your Start Menu.

- If the ETH32 device is on the same network as your PC, you can click the Detect button to populate the Combo Box with the IP address of any ETH32 device(s) on the network.
- Otherwise, you can type in the IP address you assigned to the ETH32.
- Click Connect. A message box should appear letting you know that a connection to the ETH32 was successfully made. Click OK to dismiss the message.
- You may now check and uncheck the check boxes to turn the ETH32's LEDs on and off.
- This example program may be run on more than one computer simultaneously or more than one copy (instance) on a single computer. If the state of the LEDs is modified by another instance of the example program, you may use the Refresh button to update the status of the check boxes to match the current state of the LEDs.

4. Feature Descriptions

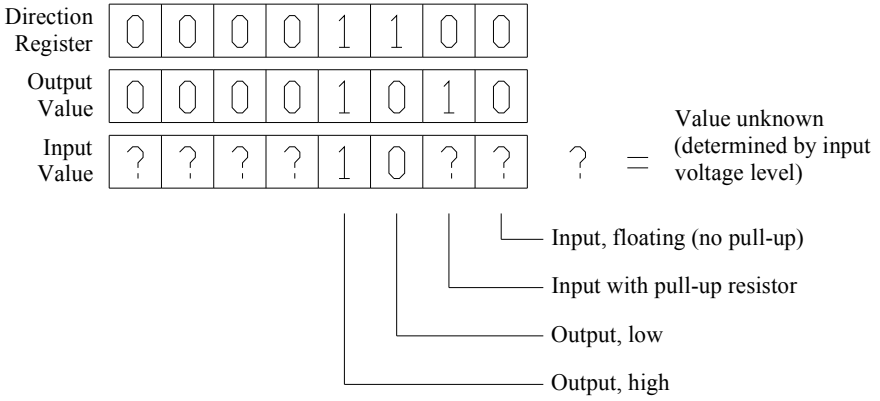
4.1. Digital I/O Ports

Digital I/O ports are one of the simplest and most frequently used means of controlling, monitoring, and interfacing with other devices. Each individual I/O line on the ETH32 can be configured as an input line or an output line. When configured as an input, a line may be configured to be pulled up internally with a pull-up resistor or float. Each digital I/O port has three registers which allow configuration and usage of the I/O lines of that port.

- Direction Register - Controls whether each line is an input or an output. This register may be read and written.
- Output Value Register - When a line is configured as an output, this register controls the output voltage (low or high). When a line is configured as an input, this controls whether an internal pull-up resistor is enabled on that line. This register may be read and written.

- Input Value Register - The value of this register indicates the state (low or high) of each line of the port. This is a read-only register.

The following diagram shows an example scenario for one of the 8-bit digital I/O ports:


Direction Register
0 0 0 0 1 1 0 0
Output Value
0 0 0 0 1 0 1 0
Input Value
? ? ? ? 1 0 ? ?
? = Value unknown (determined by input voltage level)
Input, floating (no pull-up)
Input with pull-up resistor
Output, low
Output, high

4.2. Analog Channels

The ETH32 includes the capability of reading analog input signals. There are many different available combinations of which pins are used to obtain analog readings and in what way the pin voltages are interpreted First, it is important to understand a few concepts.

4.2.1. Analog I/O Pins

There are eight analog-capable I/O pins, which are shared with the pins of digital I/O port 3. Although there are many combinations, it is helpful to remember that all analog readings originate from these eight I/O pins, the only exception being the reference channels.

4.2.2. Physical Channels

The voltage levels on the analog I/O pins can be interpreted in different ways by the ETH32 device. There are 32 different combinations offered by the ETH32 device. These are referred to as physical channels throughout the ETH32 documentation. The physical channels are grouped into the following categories:

- Single-Ended Channels measure the voltage of a pin with respect to ground.
- Differential Channels measure the voltage difference between two analog I/O pins.
- Reference Channels - measures fixed internal voltages (used for calibration purposes).

There are eight single-ended channels (one for each analog I/O pin), 22 differential channels, and two reference channels.

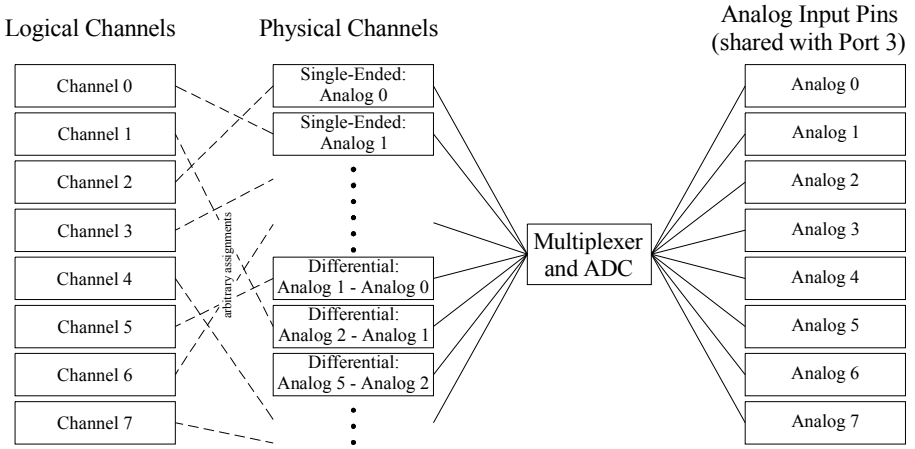
4.2.3. Logical Channels

The ETH32 uses the concept of a logical channel to allow you to specify which of the physical channels should be continuously updated within the device and potentially monitored for event thresholds (see the *Event Monitoring* section, page 11). There are eight logical channels. Each logical channel can be assigned by you to obtain its reading from any arbitrary physical channel. The logical channels are purely a firmware feature provided to improve the handling of analog readings and event monitoring - there is no physical hardware involved.

4.2.4. Analog to Digital Converter

All analog channels share a single Analog to Digital Converter (ADC). The ETH32 continuously updates the readings of the eight logical channels as quickly as the ADC and CPU time allows. Please see the *Analog Channel Refresh Rate* section on page 34 for information on the analog sample rate.

The following diagram depicts the flow of information from the I/O pins on the right side to the logical channels on the left side.



As an example, the diagram depicts the *Single-Ended Analog 1* physical channel being assigned to *Logical Channel 0*. In this case, whenever the value of logical channel 0 is read, it will provide the reading (voltage) of the Analog 1 pin with respect to ground. Please see the *Connector B Pin Assignments* section on page 24 for the location of the Analog 0 - Analog 7 pins on the I/O connector.

4.2.5. Voltage Reference

The ADC uses a reference voltage as the basis for each conversion it performs. The reference voltage provided to the ADC represents the highest possible value for a reading. In other words, the readings are scaled so that a 0V signal on an input channel will give the lowest reading (0) and an input voltage equal to the reference voltage will give the highest reading (1023 since the ADC is 10-bit). The voltage reference may be configured for any of the following options:

- Externally supplied voltage (pin 23 on Connector B)
- Internal VCC (5V)
- Internally-generated 2.56V reference voltage

CAUTION: Whatever reference voltage is selected will be internally connected to the external voltage reference pin on Connector B. Therefore, you must ensure that this pin is not connected to anything if you select one of the internal voltage reference sources.

4.3. Event Monitoring

One powerful feature of the ETH32 is its ability to locally monitor inputs and notify one or more connected PCs when certain criteria are met. This feature can, in many cases, eliminate the need for polling the ETH32 device. Two significant advantages of this approach are the increased sampling rate (the ability to catch shorter events) and the decreased network traffic that would otherwise be used for polling. In many cases, this approach also provides a more convenient and intuitive structure for programming within your PC applications. If there is ever a situation where polling is still preferred, that approach can also be implemented, either in place of or along with event monitoring.

As an example, one very appropriate use of the event monitoring capability where polling would otherwise be used is the scenario of monitoring a pushbutton. Event monitoring allows your PC application to be automatically notified each time the pushbutton is either pressed or released.

Event monitoring capabilities are included for digital I/O ports 0-3 (32 I/O lines), the analog channels, and the digital counters. Any or all of these events may be simultaneously enabled at any given time. Please see the ETH32 API documentation for a complete explanation of the available events for digital inputs, analog inputs, and counters.

4.4. Pulse Width Modulation (PWM) Outputs

Two of the I/O pins can function as Pulse Width Modulation (PWM) output signals. These allow a configurable square-wave output signal to be generated. PWM outputs are often used for speed control as well as efficiently controlling LEDs, relays, and other loads. Please see the *Pulse Width Modulation Outputs* section on page 36 for PWM output specifications. Information on controlling the PWM outputs can be found in the ETH32 API documentation.

4.5. Digital Counters

The ETH32 contains two digital hardware counters. Two of the ETH32 I/O pins can function as inputs to the counters, thus providing the ability to count logic pulses on those pins. When enabled, each

counter can be configured to increment on either the rising or falling edge of its associated I/O pin. One of the two counters is 8-bit and the other is 16-bit.

4.6. Built-in LEDs

The ETH32 device includes two user-controllable LEDs. The LEDs provide a quick and easy way to display a status condition, error, or some other condition your application needs to display.

4.7. EEPROM Storage

The ETH32 provides 256 bytes of EEPROM memory on the microcontroller which you can use for any data that you wish to store. The contents of the memory are preserved through power losses and reset, so it is perfect for storing device identification, calibration data, or other non-volatile data. EEPROM storage functionality is included with firmware v3.000 and greater.

4.8. Power Up Defaults

When the ETH32 device is initially powered up or reset, all I/O lines start out as inputs, and most features start out disabled. This helps to prevent hardware conflicts with other devices that may be connected to the ETH32. At power up or reset, the device is configured as follows:

- All digital I/O ports are configured as inputs.
- The output value registers of all digital I/O ports are set to zero.
- The LEDs are turned off.
- The Analog to Digital Converter is disabled.
- The analog voltage reference is configured to be the external reference pin.
- All events are disabled for all connections.
- Both counters are disabled and values set to zero.
- PWM channels are disabled.

4.9. TCP/IP Connections

The PC communicates with the ETH32 using the TCP/IP protocol. TCP/IP is a standard, robust, time-tested protocol used extensively on the Internet as well as local area networks. TCP/IP provides reliable delivery of data by automatically verifying the integrity of all data packets and buffering and retransmitting data in the event it becomes corrupted or lost.

The ETH32 supports multiple simultaneous TCP/IP connections from one or more computers. This allows a great deal of flexibility by allowing more than one application and more than one computer to monitor and/or control the ETH32 at one time.

Each connection is allowed to individually select the event notifications it should receive; this feature prevents unnecessary usage of network bandwidth and other resources.

4.9.1. DHCP Support

As of firmware version 3.000, the ETH32 includes support for obtaining an IP address, Network mask, and Gateway IP address settings from a DHCP server. DHCP support can be disabled to use a static IP address.

Unless your DHCP server is configured to provide a specific IP address to an ETH32, you will need to determine which IP address has been assigned in order to connect to it with the applications you develop. This can be done manually by detecting ETH32 devices with the ETH32 Configuration utility as discussed later. Or, ETH32 devices can be detected programmatically in your software using the ETH32 API, v2.00 or later. In either case, in order for a PC to detect it, the ETH32 must be on the same physical subnet as the PC.

Note that in many cases it is probably simplest to configure the ETH32 with static settings instead of using DHCP, but both ways are available in order to best suit your needs.

5. Software Installation

The CD included with the ETH32 includes several useful resources:

- ETH32 API library used for developing PC applications; required for running the example programs.
- Documentation.
- ETH32 network configuration utility.
- Example PC applications, including source code.

5.1. Windows Installation

Windows users are encouraged to use the automatic installation program to install all of the available components on their system. To do this, simply run the `eth32_install.exe` file located on the ETH32 product CD.

For those who may be wondering, the only file that is installed into your system directory (e.g. C:\Windows\System32) is the core ETH32 API library, `eth32api.dll`. This is the only run-time file required for applications that use the ETH32 API unless the applications are created with a Microsoft .NET language, in which case the `Eth32.dll` .NET assembly is also required.

5.2. Linux Installation

Linux users are encouraged to run the install.sh script to install the ETH32 API onto their system. To do this, follow these simple steps:

- Log in as root.
- Depending on your system configuration, you may need to mount the CD.
- Run the following commands:
  `cd /`*(path to cd drive)*`/api/linux`
  `./install.sh`

This script does not copy any example programs or documentation onto your system (they are available directly from the CD). For your information, it performs these tasks:

- Copies the ETH32 API libraries (`libeth32.*`) into `/usr/lib/`
- Configures the symbolic links to the shared library; these links are required for proper compile-time and run-time linking.
- Copies the `eth32.h` header file into `/usr/include/`

6. Network Connections and Configuration

6.1. Connecting the ETH32 to the Network

The ETH32 connects to an Ethernet network just like any PC or network-connected printer. The Ethernet port uses 10Base-T signaling, which means it must be connected to a 10Base-T or compatible (for example 10/100/1000) port of your Ethernet hub or switch.

Like any Ethernet device, the ETH32 may also be connected directly to a PC's Ethernet card using a "crossover cable". In a case where only two devices are required (such as a PC and a ETH32 device), a crossover cable eliminates the need for a hub or switch.

6.2. Configuring the ETH32

Before beginning to use the ETH32 device, you must configure its IP address and other network settings with values that are appropriate for your network. These settings are configured over the Ethernet connection using the included utility program. On a Windows computer, the configuration utility program will be available on the Start Menu at *Winford ETH32 ➔ ETH32 Network Configuration* if it was selected when running the installation program.

6.2.1. Default Network Configuration

The factory-default network configuration settings for the ETH32 are as follows:

- DHCP: Enabled (as of firmware v3.000)

The static IP settings stored in the ETH32 have factory-default settings as follows, but note that these settings do not take effect as long as DHCP remains enabled:

- IP Address: 192.168.1.100
- Subnet Mask: 255.255.255.0
- Gateway IP: 192.168.1.1

If you need to disable DHCP and configure static IP settings, you will need to configure the device as described below.

6.2.2. Network / Firewall Requirements

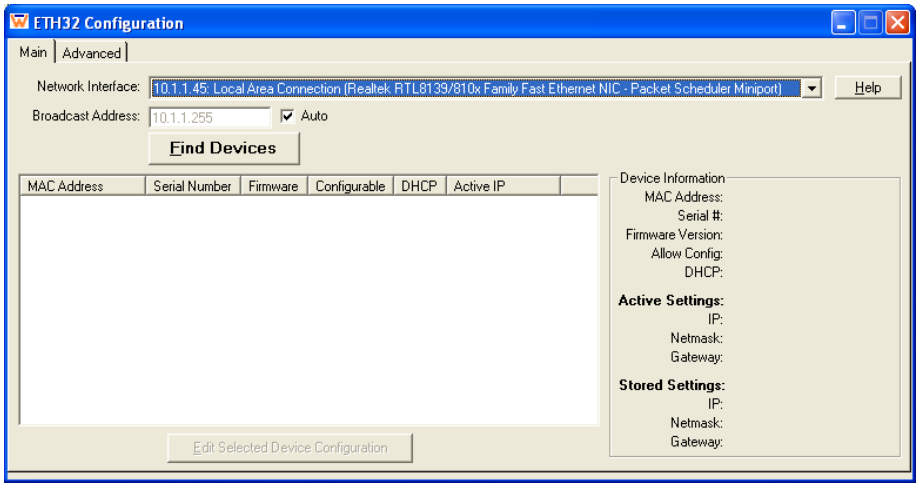
The configuration utility must be run on a PC that is on the same network segment as the ETH32 device. In other words, there may be Ethernet hubs and/or switches in between the PC and the ETH32, but there must not be any routers in between. This is because the configuration utility uses UDP broadcast packets to detect and configure the ETH32 device(s), and broadcast packets are not forwarded by routers. UDP broadcast packets are used because they allow the utility to communicate with ***any*** ETH32 device on the same network segment, regardless of whether its currently configured IP address is appropriate for the network. Note that this condition only applies to the configuration utility. During normal usage, there may be any number of routers between the PC and the ETH32. In fact, the ETH32 device could be half way around the world!

If the PC being used for configuration has a firewall enabled (whether it is built into Windows, or add-on) it *may* need to be disabled during the configuration of the ETH32. The firewall may reject configuration packets from the ETH32 if it is configured with and using an IP address that is not expected on the local network. If the ETH32 is not able to be found by the utility, be sure to disable the firewall. Once the ETH32 has been initially configured with an appropriate IP address, future configurations should not require the firewall to be disabled.

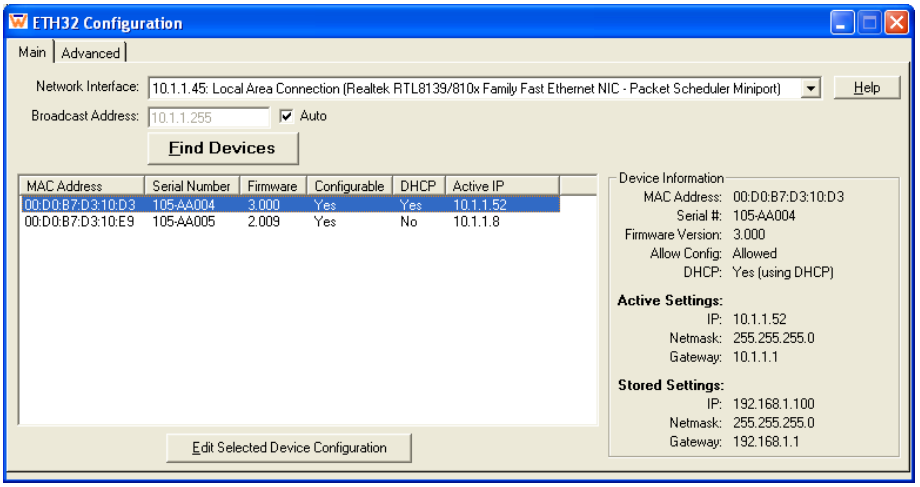
6.2.3. Configuration Utility

The ETH32 network configuration utility can be used by following these steps:

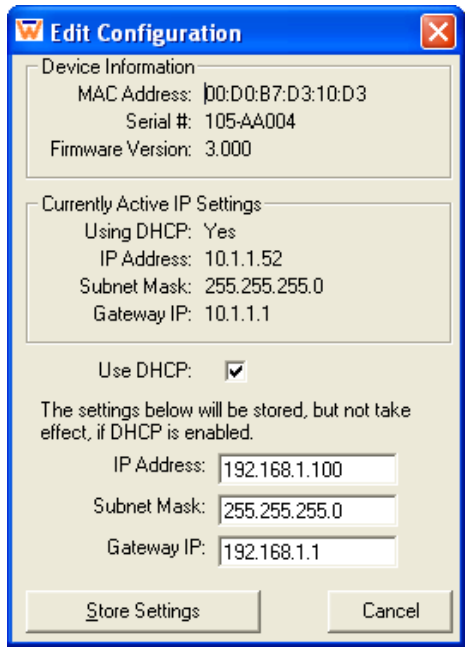
- Set the ALLOW CONFIG switch on the ETH32 device to the Y position. Otherwise, the ETH32 will not allow its configuration settings to be modified.
- Launch the configuration utility by selecting *Winford ETH32* ➔ *ETH32 Network Configuration* on the Start Menu. A window similar to this will be shown:



- Choose the network interface of your PC to which the ETH32 is connected. The appropriate broadcast address for that interface will be automatically determined and shown in the Broadcast Address box. If your PC/network has an unusual configuration that requires you to manually determine the broadcast address to which queries will be sent, you may un-check the “Auto” check box and manually enter a broadcast address. On many PCs, 255.255.255.255 will broadcast out on all network interfaces.
- Click the "Find Devices" button.
- After a few seconds, any ETH32 devices on the same network segment should be detected and displayed in the table, similar to the following:



- Examine the table and select the ETH32 which you would like to configure. Each device is identified by a unique MAC address and serial number (S/N). Both of these numbers are printed on the back of the ETH32 device.
- Click on the device you would like to configure and then click the "Edit Selected Device Configuration" button. A window similar to this will be shown:



- Modify the settings as appropriate for your network. A full explanation of these settings is beyond the scope of this manual, but brief explanations are provided here.
  - Use DHCP - If this box is checked, the ETH32 will obtain an IP address automatically using the Dynamic Host Configuration Protocol (DHCP), and the IP Address and other settings that are stored in the ETH32 will not be used. A DHCP server must be present on your network for this to work. DHCP support is included in ETH32 firmware v3.000 and greater.
  - IP Address - Enter an IP address that is valid and available on this segment of your network. Ask your network administrator for an available IP address. If you are the network administrator, you should choose an IP that is not being used by any other device and is in the same subnet as your PC. For example, if your PC's Ethernet card has the IP address of 192.168.1.1 and a subnet mask of 255.255.255.0, any available IP from 192.168.1.2 through 192.168.1.254 would be acceptable. The lowest address in a particular network (e.g. 192.168.1.0) is reserved and cannot be assigned to the ETH32 or any other device. The highest address (e.g. 192.168.1.255) is reserved as a broadcast address.
  - Subnet Mask - This setting indicates which IP addresses are local and which require going through the gateway. If you are unsure, this setting can typically be copied from the configuration of PCs on the same network segment. A setting of 255.255.255.0 is common.
  - Gateway IP - This is the IP address of the router that should be used to send data outside of the local network segment. If you are unsure, this setting can typically be copied from the configuration of PCs on the same network segment. If the ETH32 is being used on a network without a router, a value of 0.0.0.0 may be entered.
- After you have entered the new network configuration settings, click the "Store Settings" button to save the settings to the ETH32. A message box will be shown confirming that the settings have been successfully stored.

7. Developing PC Applications

The ETH32 device is controlled and utilized by PC applications that you develop and execute. The versatility of the ETH32, coupled with the almost limitless control and response possibilities of a PC application, creates a very powerful combination.

The ETH32 API provided by Winford Engineering gives your application a high-level interface to the functionality of the ETH32. The API can be used from the following programming environments or compilers:

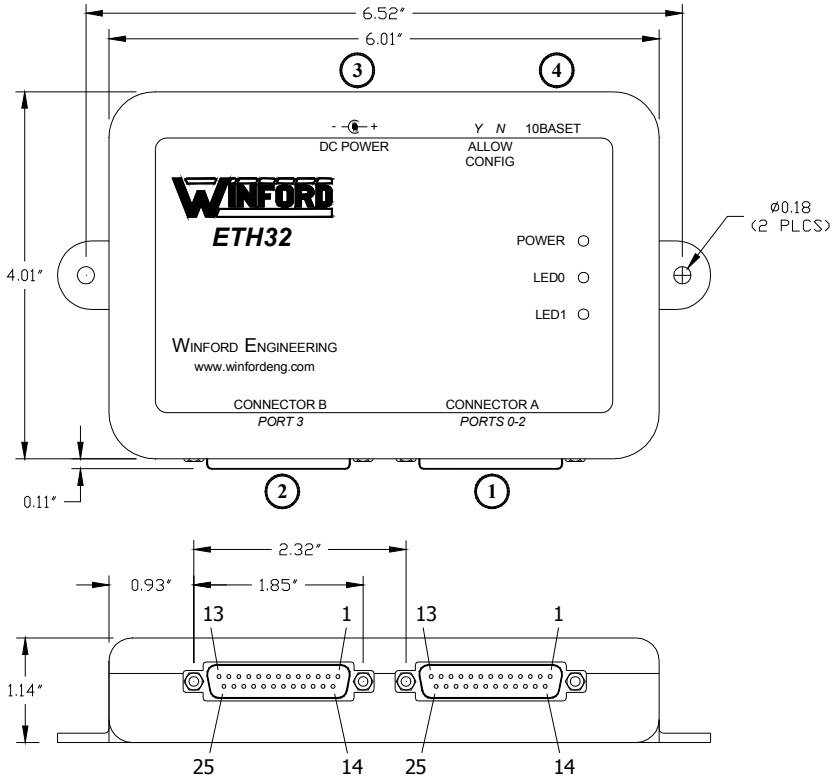
- Microsoft Visual Basic 6
- Microsoft Visual Basic .NET
- Microsoft Visual C# .NET
- Microsoft Visual C++ .NET
- Microsoft Visual Studio C++ 4 or later
- Borland C++
- Embarcadero C++ Builder
- MinGW GCC
- GNU GCC

If a language or platform is used which is not directly supported by the ETH32 API, it is also possible to control the ETH32 by directly communicating with its TCP/IP port, bypassing the API. In other words, your application can perform the tasks that the API would normally perform, if so desired. Details of the commands and responses are provided in the "ETH32 Protocol Reference" document.

Detailed documentation of the ETH32 API is included on the ETH32 product CD. The documentation includes topical overviews, requirements for the different supported languages, and a complete function/member reference for the supported languages.

8. Hardware Information

8.1. Dimensions and Connector Locations

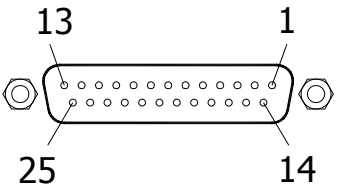


1. **Connector A: Ports 0, 1, and 2**
2. **Connector B: Port 3, miscellaneous signals**
3. **DC Power Jack**
4. **RJ45 Ethernet Jack**

8.2. Connector Pinouts

8.2.1. DB25 Connectors

Both DB25 connectors are female connectors. The DB25 female connector pins are numbered as shown on the following diagram:



Connector A Pin Assignments

Pin	*Primary Function*	*Alternate Function*
1	Port 0, bit 0	
2	Port 0, bit 1	
3	Port 0, bit 2	
4	Port 0, bit 3	
5	Port 0, bit 4	
6	Port 0, bit 5	
7	Port 0, bit 6	Counter 0
8	Port 0, bit 7	Counter 1
9	Port 1, bit 0	
10	Port 1, bit 1	
11	Port 1, bit 2	
12	Port 1, bit 3	
13	Port 1, bit 4	
14	Port 1, bit 5	
15	Port 1, bit 6	
16	Port 1, bit 7	
17	Port 2, bit 0	
18	Port 2, bit 1	
19	Port 2, bit 2	
20	Port 2, bit 3	
21	Port 2, bit 4	PWM 0 Output
22	Port 2, bit 5	PWM 1 Output
23	Port 2, bit 6	
24	Port 2, bit 7	
25	Ground	

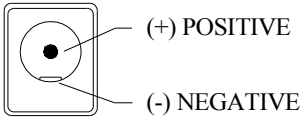
Connector B Pin Assignments

Pin	*Primary Function*	*Alternate Function*
1	Port 3, bit 0	Analog 0
2	Port 3, bit 1	Analog 1
3	Port 3, bit 2	Analog 2
4	Port 3, bit 3	Analog 3
5	Port 3, bit 4	Analog 4
6	Port 3, bit 5	Analog 5
7	Port 3, bit 6	Analog 6
8	Port 3, bit 7	Analog 7
9	Port 4, bit 0	
10	Port 5, bit 0	
11	Reset (Active Low)	
12	V_{CC}	
13	V_{CC}	
14	Ground	
15	Ground	
16	Ground	
17	Ground	
18	Ground	
19	Ground	
20	Ground	
21	Ground	
22	Ground	
23	Analog Voltage Ref.	
24	Power in (unregulated)	
25	Ground	

The Reset signal (pin 11) is internally pulled high to allow the ETH32 to function without any connection to the Reset line. Pulling the Reset line low will reset the ETH32 and hold it in reset for as long as the line remains low.

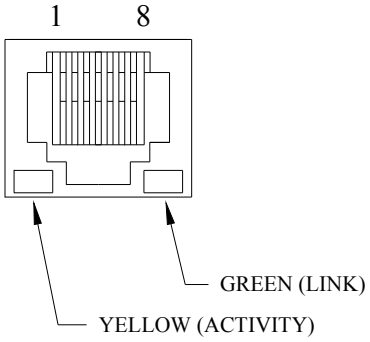
8.2.2. DC Power Jack

The DC power jack accepts connectors with a 2.1mm inside diameter and 5.5mm outside diameter. The jack requires a center-positive supply. See the Specifications section starting on page 29 for the acceptable voltage range.



8.2.3. RJ45 Ethernet Jack

The ETH32 supports a 10Base-T Ethernet connection to the RJ45 jack. The jack is wired in the standard way for Ethernet network devices and requires no special cables. The jack includes two LEDs which indicate network link status and network activity. The green LED lights when the link is established. The yellow LED lights when network activity is occurring. It will light brightly when both transmit and receive traffic is occurring and dimly when only one or the other is occurring.



RJ45 Jack Pin Assignments

Pin	*Function*
1	TX+: Transmit Data +
2	TX-: Transmit Data -
3	RX+: Receive Data +
4	Not connected
5	Not connected
6	RX-: Receive Data -
7	Not connected
8	Not connected

8.3. Power Options

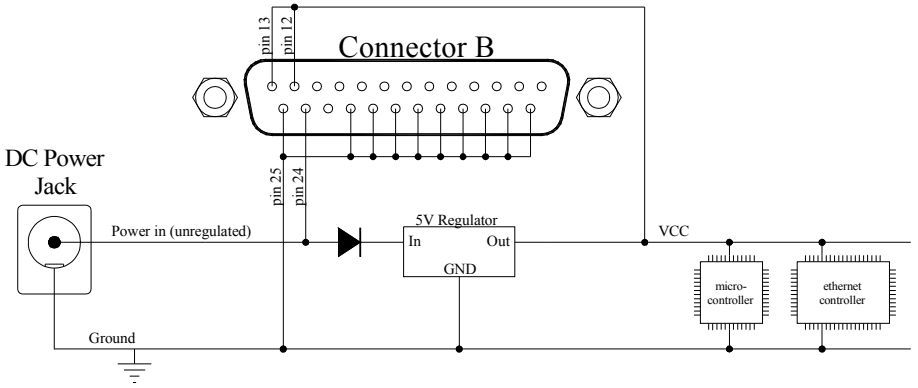
The ETH32 device may be powered from one of several methods:

- The DC power jack
  This accepts unregulated DC power. It is internally regulated and protected against accidental reverse polarity.
- DB25 unregulated power pin (Connector B, pin 24)
  This is directly connected to the DC power jack. It is therefore also internally regulated and protected against accidental reverse polarity.
- DB25 V_{CC} pins (Connector B, pins 12 and 13)
  The V_{CC} pins are connected to the output side of the internal voltage regulator, and may be used to provide power to the ETH32, or to power low-current auxiliary devices.
  - **Auxiliary Power Output:** If the board is powered through one of the above options, the V_{CC} pins may be used to provide low-current 5V power for auxiliary devices.

- **Power Input:** Alternately, the ETH32 may be powered through the V_{CC} pins. However, there is no voltage regulation or reverse polarity protection. Clean, regulated 5V power must be provided and extra care must be taken to connect power in the right polarity. Failure to meet these requirements will permanently damage the device.

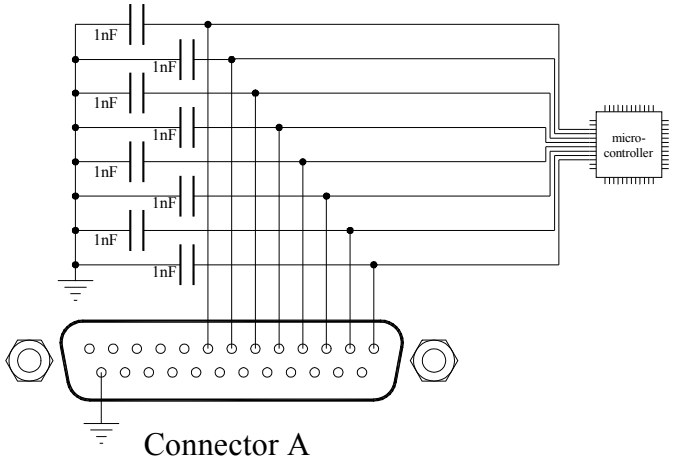
Power should be supplied by only one method at a time.

The ETH32's internal power connections are summarized in the following schematic.



8.4. I/O Ports

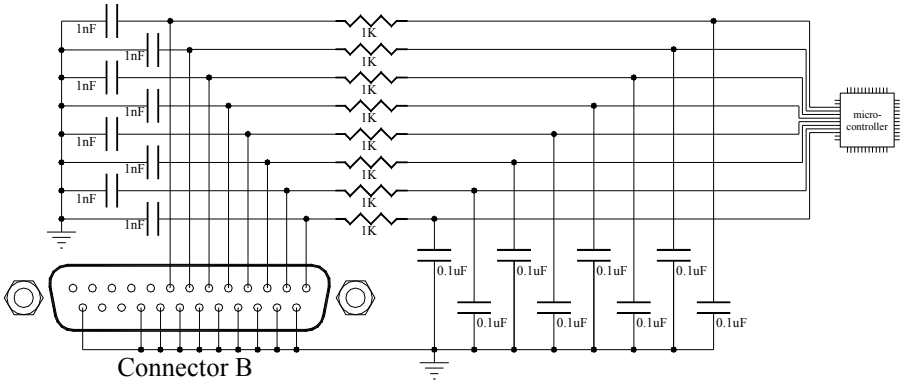
The ETH32 has four 8-bit I/O ports (ports 0, 1, 2, and 3) and two 1-bit I/O ports (Ports 4 and 5). For a**ll I/O pins**, the ETH32 includes internal 1nF capacitors between the connector pin and ground near the DB25 connector in order to reduce the chances of electrical noise causing erratic operation.



Connector A

(PORT 0 SHOWN - PORTS 1,2,4,5 SAME)

On all ports except port 3, the DB25 connector pins are connected directly to the microcontroller in the ETH32. Because port 3 is also used for analog inputs, the ETH32 includes simple internal 1K / 0.1uF R/C filters on these lines in addition to the 1nF capacitors. The following diagram shows the configuration of port 3.



Connector B

(PORT 3 ONLY)

9. Specifications

9.1. Absolute Maximum Ratings

Storage Temperature ..... -30°C to 70°C

Ambient Operating Temperature ..... 0°C to 55°C [1]

Operating Humidity.....5% to 95% non-condensing

Auxiliary Power Output Current ..... 250 mA

Voltage on any I/O pin with
respect to Ground ..... -0.5V to V_{CC}+0.5V

Voltage on External Analog Voltage
Reference with respect to Ground .....-0.5V to V_{CC}+0.5V

Voltage on Reset with respect to Ground ..... -0.5V to +13.0V

Unregulated Power Supply Voltage ..... 25V [2]

Maximum V_{CC} Operating Voltage
(if device is powered through V_{CC} pins) 6.0V

DC Current per I/O pin .....40.0 mA

Maximum I/O Port Sink Current:
- Sum of all I/O pins .....390 mA
- Sum of Port 0 pins .....85 mA
- Sum of Ports 1, 2, 4, and 5 pins .....100 mA
- Sum of Port 3 pins .....100 mA

Maximum I/O Port Source Current:
- Sum of all I/O pins .....400 mA
- Sum of Port 0 pins .....95 mA
- Sum of Ports 1, 2, 4, and 5 pins .....100 mA
- Sum of Port 3 pins .....100 mA

Notes:

1. The specified operating temperature is based on the following conditions:
   - 7.5VDC unregulated power input
   - 250 mA auxiliary power output current
   - Every I/O pin driving a 1kΩ load

   Variations of this setup could adjust this temperature range. For unusual requirements or environments, please contact Winford Engineering.
2. Voltages above 7.5V could decrease the maximum operating temperature range.

Warning:

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not recommended or implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

9.2. Electrical Characteristics

Parameter	*Min*	*Typical*	*Max*	*Unit*
Unregulated Power Input	7.0		20	V
V_{CC} Voltage (voltage regulator output when power is supplied to unregulated input)	4.90	5.00	5.10	V
Acceptable V_{CC} Operating Voltage (when power is supplied to V_{CC} pins)	4.75	5.00	5.25	V
Input Low Voltage	-0.5		$0.2 \times V_{CC}$ (1)	V
Input High Voltage	$0.6 \times V_{CC}$ (2)		$V_{CC} + 0.5$	V
Reset Pin Low Voltage	-0.5		$0.2 \times V_{CC}$ (1)	V
Reset Pin High Voltage	$0.85 \times V_{CC}$ (2)		$V_{CC} + 0.5$	V
Output Low Voltage (Condition: I_{OL}=20 mA, V_{CC}=5V)			0.7	V

Parameter	*Min*	*Typical*	*Max*	*Unit*
Output High Voltage (Condition: I_{OH}=-20 mA, V_{CC}=5V)	4.2			V
Internal I/O Pin Pull-up Resistors (when enabled)	20		50	kΩ
Idle current		67 (3)		mA

Notes:

1. "Max" means the highest value where the pin is guaranteed to be read as low.
2. "Min" means the lowest value where the pin is guaranteed to be read as high.
3. This is the typical current drawn by an ETH32 device at 25°C which has no outputs enabled, both user LEDs off, and has no network traffic.

9.3. Performance Characteristics

There are a number of performance characteristics of the ETH32 device which are affected by other tasks the ETH32 is performing. This section provides some information about the typical performance level the ETH32 achieves in various scenarios. There are many aspects that affect these performance characteristics, including network traffic, hub/switch performance, the number of connections to the ETH32, the number of enabled events, and the operating system and speed of the PC. Therefore, these numbers should not be considered as guaranteed, but they can still provide a rough idea of ETH32 performance. Any of these characteristics are subject to change with future firmware versions.

9.3.1. Basic Test Scenario

Unless otherwise specified, the characteristics provided below were measured within the following test scenario:

- A quiet (low-traffic) Ethernet network
- Ethernet hub or switch between ETH32 and PC
- A single active connection to the ETH32

- ETH32 in power-up default state (No events enabled, most features disabled)
- ETH32 firmware version 3.000
- Where a PC is involved, 2GHZ dual-core PC running Windows XP, mostly idle. Any test applications utilized the ETH32 API.

9.3.2. Digital I/O Port Reads

For the purposes of this measurement, a digital I/O port read is defined as retrieving the current value of any of the digital I/O ports of the ETH32 device. The PC application performs only one I/O read at a time. In other words, the application waits until the port's value has been returned before beginning another I/O read operation.

Rate of digital I/O reads, on average: **1,200** reads per second

9.3.3. Digital I/O Port Writes

Digital I/O write operations can, on average, be performed faster than I/O reads since there is no need to wait for data to be returned by the ETH32. Many "write commands" to the ETH32 are automatically combined into larger TCP/IP packets by the PC's TCP/IP stack, greatly reducing overhead. Therefore, the resulting write operations in this test were not evenly spaced, but rather a burst followed by a short pause and another burst and so on.

Rate of digital I/O writes, on average: **24,000** writes per second

9.3.4. Digital I/O Port Write Burst Rate

As discussed above, when sending a stream of output commands, the TCP/IP stack of the PC automatically groups output commands into packets. The ETH32 will process all commands in each packet together, as quickly as possible, followed by a short gap until the next packet is received. This test measures the maximum digital output burst rate that occurs as the ETH32 processes a single packet containing multiple output commands.

Digital Output Burst Rate: **72.4 kHz**
(13.8 microseconds between writes)

9.3.5. Command Latency

Latency is the time from when an operation is initiated on the PC to when it takes effect on the ETH32. This test was performed by measuring the typical latency time of a single digital I/O port write, that is, the time from when the write operation was initiated on the PC to when the voltage of the I/O pins changed on the ETH32.

Command latency, typical: **0.37 ms**

9.3.6. Digital Events

The digital I/O ports are continuously polled internally by the ETH32 as quickly as CPU time allows. These characteristics describe the rate at which the ETH32 is able to poll the I/O ports in various scenarios.

Test scenario as follows:

- One connection active
- One port event enabled

Digital I/O event polling rate: **17.5 kHz** average

Test scenario as follows:

- ADC enabled
- Five connections active
- All connections have port events enabled on ports 0-3 and analog events enabled on all channels of both analog event banks

Digital I/O event polling rate: **6.0 kHz** average

Note that the above monitoring rates are on average, meaning the amount of time between each sample can vary. With no other activity, the amount of time between each sample is quite consistent. However, incoming and outgoing data will introduce delays, as this near-worst-case test scenario shows:

- ADC enabled
- Five connections active

- All connections have port events enabled on ports 0-3 and analog events enabled on all channels of both analog event banks
- One connection performing port write operations as fast as possible (in this scenario, about 8,200 per second). This causes events to be fired on all connections (although many are discarded due to lack of buffer space).

In this scenario, the event polling rate was 2.4 kHz on average. If all port reads were evenly spaced, that would mean the period between each time the ETH32 polled the I/O ports was 0.4 ms (1 / 2.4 kHz). **However, this is not the case**, due to the CPU time required for processing incoming and outgoing packets. In this scenario, the period between polls reached up to **1.4 ms**.

9.3.7. Analog Channel Refresh Rate

Since the ETH32 provides eight logical analog channels, all supporting event monitoring, but has only a single Analog to Digital Converter, the ETH32 must continuously cycle through and update the eight logical channel readings. The refresh rate given here describes how quickly the ETH32 performs each update cycle and therefore the refresh rate of each logical channel. This test was performed within the "Basic Test Scenario" described in section 9.3.1 above, with the exception of:

- ADC enabled
- Analog events enabled on all channels of both analog event banks

Analog Channel Refresh Rate, typical: **470 Hz** average

Each time an analog channel reading is refreshed, the ETH32 checks the reading to determine whether any analog events should be fired. Therefore, the above refresh rate is also the average analog event polling frequency in this scenario.

9.3.8. DHCP Overhead

When DHCP is enabled, there will be a very small amount of additional CPU overhead. The worst-case scenario occurs at the time a lease needs to be renewed, at which point an additional delay of about **0.6ms** will occur, typically just once per lease renewal. If it happened to come at a worst-case moment, this delay could be in addition to the delay/timing numbers provided above. How frequently lease renewal occurs depends on the DHCP server configuration, but can range from every few minutes to once a week, or even longer.

9.3.9. User EEPROM Writing

Storing data into the user-accessible EEPROM memory is a relatively slow process.

Time to write one EEPROM byte, typical: **9 ms**

The whole process of writing all 256 bytes will therefore take about 2.3 seconds. This will temporarily interrupt event monitoring (although not for the entire 2.3 seconds at once) while the write is taking place.

9.4. TCP/IP Connections

The ETH32 supports **five (5)** simultaneous TCP/IP connections.

9.5. Analog to Digital Converter

The analog to digital converter (ADC) is contained within the microcontroller used on the ETH32. It provides these specifications:

- 10-bit Resolution
- 0.75 LSB (least significant bit) Integral Non-linearity
- ±1.5 LSB Absolute Accuracy
- For differential channels, if 1x or 10x gain is used, 8-bit resolution can be expected. If 200x gain is used, 7-bit resolution can be expected.

9.6. Counters

- Counter 0: 16-bit counter (values from 0 to 65,535)
- Counter 1: 8-bit counter (values from 0 to 255)

Both counters are hardware counters contained within the microcontroller used on the ETH32. Therefore, the ability of the counters to detect rising/falling edges is not affected by network traffic, the number of active connections, or other resource usage.

9.7. Pulse Width Modulation Outputs

- Two PWM Outputs
- Generated from 16-bit counter clocked at 2MHZ
- Base frequency range: 30.5 Hz (65,535 counts) - 40 kHz (49 counts)
- Duty period of each PWM output is independently configurable. Base frequency is shared between outputs.
- Duty period resolution varies depending on base frequency. At the lowest frequency (30.5 Hz), there are 65,536 possible duty period settings. At the highest frequency (40 kHz), there are 50 possible duty period settings.

10. Appendix: ETH32 Accessories

A number of useful accessories are available for the ETH32. These can be seen and purchased on-line at: www.winford.com

10.1. Power Supply

Winford Engineering's model WWD075-12 wall power supply is a suitable power supply for powering the ETH32 through its DC power jack.

http://www.winford.com/products/wwd.php

10.2. Breakout Boards

Whether in the prototyping phase or permanent installation, it can be very convenient to have all of the ETH32 pins available in the form of

screw terminals. Winford Engineering offers several varieties of breakout boards for this purpose, including a specialized board designed specifically for the ETH32. The various options are shown below.

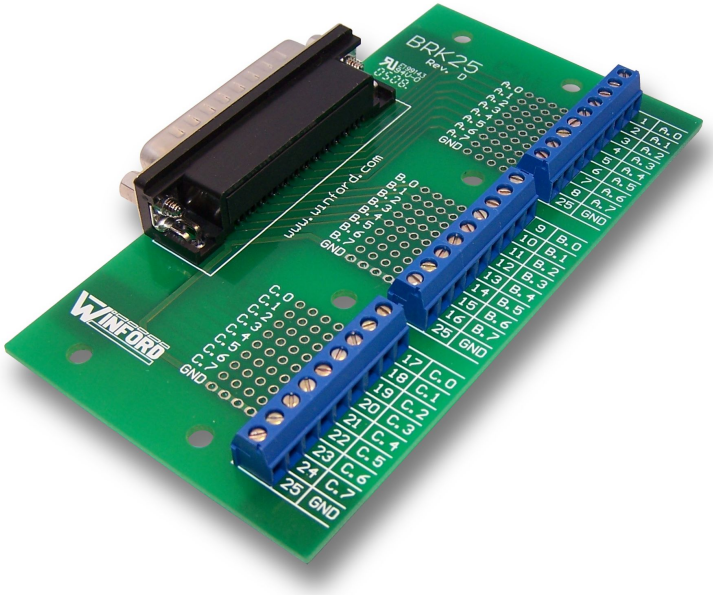
10.2.1. BRKW01 Deluxe Breakout Board for ETH32

The BRKW01 breakout board is designed specifically for the ETH32 and brings all pins from both connectors out to convenient screw terminals. In addition, many extra screw terminal ground connections are available. Handy reference information on various ETH32 I/O pins is printed directly onto the board, saving time and reducing wiring errors.

http://www.winford.com/products/brkw01.php

10.2.2. General Purpose DB25 Breakout Boards

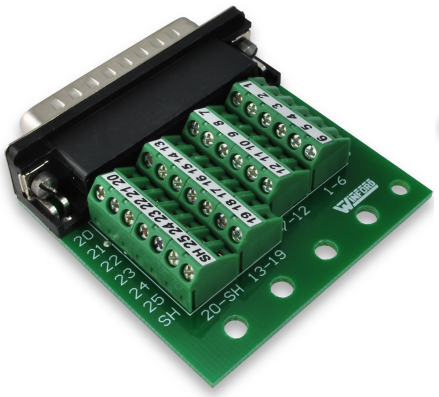
These general purpose breakout boards bring each pin of a DB25 connector out to screw terminals. They are available with male (Part# BRK25M) or female (Part# BRK25F) connectors installed and with either rubber feet or DIN rail mounting clips. Since the ETH32 has DB25 female connectors, the DB25 male breakout board (BRK25M) would typically be used so that a DB25 male-female cable (such as Winford Part# EXT25-6) can be used to connect them.



http://www.winford.com/products/brk25.php

10.2.3. Slim Footprint DB25 Breakout Boards

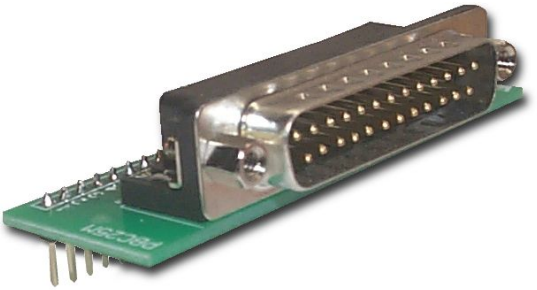
These slim form factor DB25 breakout boards offer the option of mounting a general-purpose breakout board directly to the ETH32 without the use of a connecting cable. They are available with both male (BRKSD25M) and female (BRKSD25F) connectors. They can be purchased with jack screws or captive screws installed.



http://www.winford.com/products/brksd25.php

10.3. Breadboard Adapters

Winford Engineering also offers breadboard adapters that allow a DB25 connection to be easily brought into a solderless breadboard. These are Winford Engineering product numbers PBC25F and PBC25M.



http://www.winford.com/products/pbc25.php

10.4. Cables, Connectors, Adapters

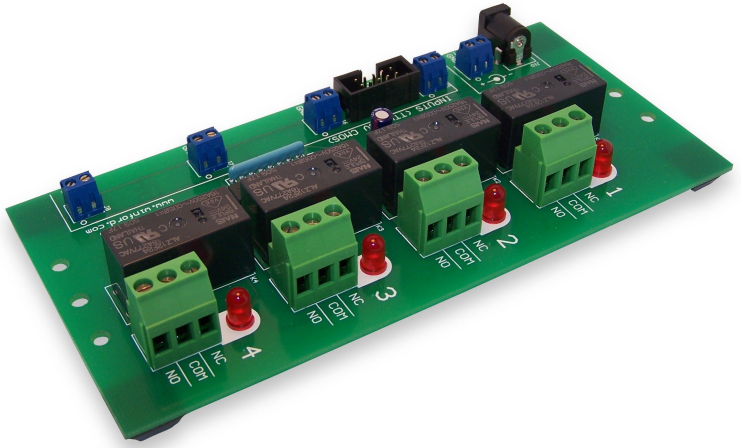
Winford Engineering offers:

- Several varieties of DB25 extension cables (EXT25-*x*).

- Network cables (NET*x*, etc.)
- DB25 solder cup connectors (CNS25*x*)
- DB25 IDC ribbon cable connectors (CNI25*x*)
- DB25 gender changers (MGC25*x*)

10.5. Relay Boards

If your project requires switching a load (for example, a 120 VAC device), you may be interested in Winford Engineering's relay boards. The relay boards include all the necessary circuitry to drive a relay with the logic-level signals provided by the ETH32. Two-channel and four-channel relay boards are currently available with several different relay coil voltage options. The root part numbers are RLY102 and RLY104, respectively.



http://www.winford.com/products/cat_rly.php

10.6. Wireless Adapter

If you need to communicate with the ETH32 where network cabling is currently not available, a wireless network adapter may be an appropriate solution. Winford Engineering currently does not sell such adapters, but there are several available on the marketplace. These act as a bridge between a standard wired Ethernet network and a wireless Ethernet network. Some examples of these products include:

- D-Link DWL-G820
- Linksys WET11